CHAPITRE I

INTRODUCTION

I.1 Préambule

Le but de ce chapitre est de présenter brièvement au lecteur l'éditeur de texte. Pour ce faire, nous illustrerons son fonctionnement par plusieurs exemples . la convention suivante est utilisée : ce qui est souligné est frappé au clavier, ce qui ne l'est pas est affiché par l'éditeur.

L'éditeur appelé répond (1) (NEW FILE par exemple comme ci-dessous). Dès lors nous créerons notre fichier simplement en frappant toutes les lignes qui le composent, chaque ligne se terminant par un "retour chariot".

```
NEW FILE :

      1.00 = VØICI UN EXEMPLE D'UTILSATIØN DE
      2.00 = L'EDITEUR DE TEXTE. BEAUCØUP
      3.00 = D'EXEMPLES NØUS PERMETTRØNT D'APPRENDRE
      4.00 = RAPIDEMENT ET FACILEMENT SES CARACTERISTIQUES
      5.00 = LES LIGNES SUIVANTES N'ØNT AUCUN SENS :
      6.00 = ABCDEFGHIJKL
      7.00 = AAAAAAAA
      8.00 = TEST 1234
      9.00 = L'EDITEUR EST AMUSANT A UTILISER !
     10.00 = BBBBBBBB
     11.00 =
     12.00 = NØUS VØICI A LA FIN DE CE FICHIER
     13.00 = AU MØINS PØUR L'INSTANT
     14.00 = #
     13.00 = AU MØINS PØUR L'INSTANT
#
```

(1) cf Ch.III

> Attention ! Ø représente la lettre O.
> 0 représente le chiffre zéro.

Vous remarquerez qu'il a fallu frapper un dièse (#) en première colonne pour sortir du mode saisie. A ce moment le système a récrit la dernière ligne et affiché le dièse. L'éditeur est prêt à accepter des commandes. A tout instant , quand on frappe des caractères dans l'éditeur, deux caractères ont un sens spécial :

BS (BACKSPACE ou CØNTRØL H)	Efface le dernier caractère (retour arrière)
CAN (CØNTRØL X)	Efface toute la ligne courante

Ces commandes sont utiles, en cas de fautes de frappe, et permettent la correction immédiate. (sur GOUPIL 2 utiliser les touches ← ou CTRL H, et CTRL X du clavier.).

I.2 Identification des lignes

Un numéro est donné à chaque ligne de texte de l'éditeur et permet d'identifier la ligne. Chaque numéro de ligne est de la forme m.nn' où m est un nombre entier et n et n' des chiffres de 0 à 9. 73, 73., 73.0 et 73.00 désignent la même ligne ; de même pour 259.6 et 259.60.

Le plus grand numéro utilisable est : 9999.99

Nous noterons le repérage d'une ligne par le symbole "<ligne>". Nous utiliserons cette convention tout au long de ce document.

Un ordre indique à l'éditeur l'action à accomplir et, si besoin est , la ligne ou le bloc de lignes affecté par cet ordre.

I.3 Les commandes de l'éditeur

L'éditeur peut supprimer ou insérer des lignes de texte dans un fichier, imprimer ces mêmes lignes de texte et ainsi de suite. A chaque possibilité correspond une instruction : par exemple une instruction de suppression, une instruction d'insertion, une instruction d'impression, etc. Si nous définissons le symbole "<instruction>" pour signifier "instruction" à l'éditeur, la forme de base d'une instruction d'édition est :

"<ligne>" "<instruction>"

Par exemple, l'ordre pour afficher la ligne 12.00 est :

≠ 12 P

12.00 = NØUS VØICI A LA FIN DE CE FICHIER

ou 12 est la spécification de la ligne et P est l'instruction dans cet ordre.

I.4 L'insertion

Dans l'emploi normal du mot "insérer" nous entendons par exemple "insérer cette carte après cette autre carte". Pour utiliser l'instruction d'insertion nous précisons le N° de la ligne après laquelle la nouvelle ligne doit être insérée, suivi du Symbole I

⟨ligne⟩ I

Après la frappe de l'instruction suivie d'un "retour chariot" (↵), l'éditeur va choisir un numéro de ligne approprié et inviter à frapper la ligne en affichant ce numéro de ligne suivi d'un signe égal (=). Après que chaque ligne et le retour chariot ont été tapés, l'éditeur affiche un nouveau n° de ligne suivi d'un égal (=). Pour sortir du mode "insertion" taper un dièse (≠). Un nouvel ordre peut alors être donné à l'éditeur.

Voici quelques exemples d'insertion :

≠ 8I

8.10 = VØICI UNE LIGNE INSEREE

8.20 = AINSI QUE CELLE-CI

8.30 = ≠ 11 I

11.10 = ENCØRE UNE LIGNE INSEREE

11.20 = ≠ 6P

6.00 = ABCDEFGHIJKL

Note :

L'éditeur peut renuméroter des lignes qui suivent le texte inséré ; ceci quand assez de lignes sont insérées et que leurs numéros recoupent ceux des lignes du texte original.

I .5 La suppression

Avec cette instruction on peut supprimer une ligne ou un bloc de lignes.

Pour supprimer une ligne indiquons la <ligne> à supprimer suivie de D

<ligne> D

Lorsque vous faites le retour chariot, la ligne disparaît.

Pour supprimer plusieurs lignes, il nous faut indiquer, non seulement la première ligne à supprimer mais aussi la dernière que nous appellerons ligne cible et que nous indiquerons par <cible> Bien que l'éditeur permette plusieurs moyens de spécifier la ligne cible, voyons d'abord les deux plus simples :

1°) <cible> peut être le nombre de lignes à supprimer (en incluant la première et la dernière ligne du bloc)

2°) <cible> peut être un dièse (#) suivi du n° de la dernière ligne du bloc à supprimer.

Quelques exemples :
3 (supprimer 3 lignes)
26 (supprimer 26 lignes)
26 (supprimer jusqu'à la ligne 26.00)

La syntaxe de suppression d'un bloc est :

<ligne> D <cible>

ou <ligne> indique le N° de la première ligne
et <cible> l'étendue de la suppression.

Pour illustrer l'emploi de l'instruction de suppression supposons que nous avons un fichier comportant cinquante trois lignes numérotées de 1 à 53.

Avec les instructions :

```
# 15 D
# 24 D # 31
# 52 D 2
BØTTØM ØF FILE REACHED
#
```

Nous nous trouvons maintenant avec un fichier comportant les lignes 1 à 14, 16 à 23, 32 à 51. La première instruction a supprimé la ligne 15, la seconde les lignes 24 à 31, la troisième 2 lignes à partir de la ligne 52. Comme la dernière ligne du fichier a été supprimée, l'éditeur affiche :

FIN DU FICHIER ATTEINTE

Avant d'examiner de nouvelles instructions, nous devons étendre les définitions de <ligne> et <cible>.

I.6 La notion de ligne

Durant les opérations d'édition, l'éditeur garde trace de la ligne courante qui est généralement la dernière ligne affectée par une instruction.

Lors de l'appel du programme d'édition la ligne courante est la première ligne du fichier.

Si par exemple nous venons d'insérer trois lignes, entre les lignes 12.00 et 13.00, la ligne courante sera 12.30.

Après la suppression d'une ou plusieurs lignes, la ligne suivant immédiatement la dernière ligne supprimée devient la ligne courante (si la dernière ligne du fichier a été supprimée, c'est la nouvelle dernière ligne du fichier qui devient ligne courante).

Dans nos considérations précédentes, nous avons signalé qu'il fallait explicitement indiquer une ligne pour chaque instruction en spécifiant le numéro de la ligne intéressée.

Néanmoins, si aucune ligne particulière n'est mentionnée la ligne courante est prise par défaut.

Par exemple, en réponse à l'ordre

```
# D2
#
```

l'éditeur supprimera deux lignes à partir de la ligne courante : puisque dans notre exemple nous en étions à la ligne 6.00, les lignes 6.00 et 7.00 seront supprimées. Comme vous vous en rendrez compte à l'usage, la considération de la ligne courante en cas d'absence de <ligne> (par défaut) est extrêmement commode.

Après toutes ces manipulations notre fichier est devenu :

```
 1.00 = VØICI UN EXEMPLE D'UTILISATIØN DE
 2.00 = L'EDITEUR DE TEXTE. BEAUCØUP
 3.00 = D'EXEMPLES NØUS PERMETTRØNT D'APPRENDRE
 4.00 = RAPIDEMENT ET FACILEMENT SES CARACTERISTIQUES
 5.00 = LES LIGNES SUIVANTES N'ØNT AUCUN SENS
 8.00 = TEST 1234
 8.10 = VØICI UNE LIGNE INSEREE
 8.20 = AINSI QUE CELLE-CI
 9.00 = L'EDITEUR EST AMUSANT A UTILISER !
10.00 = BBBBBBBB
11.00 =
11.10 = ENCØRE UNE LIGNE INSEREE
12.00 = NØUS VØICI A LA FIN DE CE FICHIER
13.00 = AU MØINS PØUR L'INSTANT
```

Nous avons vu que l'on peut spécifier une ligne par son numéro ; Par défaut, nous tombons sur la ligne courante. Il existe aussi plusieurs autres moyens de spécifier une ligne ou, en d'autres termes, de déplacer le "pointeur" vers une ligne concernée avant l'éxécution de l'instruction. On peut aussi spécifier la ligne par "+n" ou "-n" (n entier) pour dire nième ligne suivante ou nième ligne précédente. Deux autres indicateurs utiles sont "∧" et " ! " qui spécifient directement la première ligne ou début du fichier et la dernière ligne ou fin du fichier. Tous ces moyens d'indiquer une ligne sont illustrés dans l'exemple ci-dessus en conjugaison avec l'instruction PRINT.

```
#_∧P
          1.00 = VØICI UN EXEMPLE D'UTILISATIØN DE
# + 3P
          4.00 = RAPIDEMENT ET FACILEMENT SES CARACTERISTIQUES
# ! P
         13.00 = AU MØINS PØUR L'INSTANT
#-2 P
         11.10 = ENCØRE UNE LIGNE INSEREE
#
```

Parfois,une partie du contenu de la ligne est connu mais pas son numéro, ni sa position par rapport à la ligne courante. Dans ce cas le caractère "analyse du contenu"de l'éditeur permet de réagir.

La syntaxe devient alors :

/ ⟨chaîne de caractères⟩/

Les // délimitent la chaîne de caractèresdont on connait l'existence dans la ligne recherchée.

Dans ce cas l'éditeur recherche, à partir de la ligne courante la prochaine ligne qui contient cette chaîne.

Exemples :

/RENARD/

L'éditeur recherche la prochaine ligne qui contient la chaîne de caractère RENARD. Si la chaîne existe la ligne qui la contient devient la ligne courante sur laquelle l'opération demandée sera exécutée. Si la chaîne n'est pas trouvée la ligne courante ne change pas et le message "NØ SUCH LINE" apparait. Notons que le séparateur n'est pas forcément un "/", ce peut être une apostrophe (') ou un point virgule (;). Par exemple :

'A/B' envoie à la prochaine ligne contenant A/B.

On peut faire précéder la chaîne désignée par le signe moins "-" pour indiquer une ligne précédente contenant la chaîne recherchée. Quelques exemples sur notre fichier vont illustrer cette méhode de recherche d'une ligne.

```
# -/RAPIDEMENT/P

 4.00 = RAPIDEMENT ET FACILEMENT SES CARACTERISTIQUES

#;123;P

 8.00 = TEST 1234

# +"FIN"P

12.00 = NØUS VØICI A LA FIN DE CE FICHIER

#
```

En résumé, une ligne peut être spécifiée par :

1) la ligne courante par défaut
2) le numéro de la ligne
3) "+N" soit la nième ligne suivant la ligne courante
4) "-N" soit la nième ligne précédant la ligne courante
5) / chaîne de caractères/ soit la première ligne contenant cette chaîne de caractèresà partir de la ligne courante
6)-/ chaîne de caractères: soit la première ligne contenant cette chaîne de caractères , rencontrée en remontant les lignes, à partir de la ligne courante.
7) "∧" pour désigner la première ligne du fichier
8) " ! " pour désigner la dernière ligne du fichier

I.6 bis - La notion de cible

Vous vous rappelez que ⟨cible⟩ est utilisée avec certaines instructions pour indiquer le nombre de lignes affectées par l'opération en question. Nous avons déjà vu qu'une cible peut être spécifiée par :

1) Un entier n (au besoin négatif) pour indiquer le nombre de lignes qui sont affectées à partir de la ligne désignée.

   Par exemple :

   # P3 pour imprimer trois lignes (à partir de la ligne courante)

2) # numéro de ligne, pour indiquer le numéro de la dernière ligne affectée par la commande

   Par exemple :

   P#6 (impression depuis la ligne courante jusqu'à la ligne n° 6 comprise). Mais nous pouvons aussi utiliser :

3) par défaut, l'ordre est effectué sur la seule ligne désignée.

4) / chaîne de caractères / pour indiquer que la commande sera exécutée depuis la ligne désignée jusqu'à la prochaine ligne contenant cette chaîne de caractère.

5) -/ chaîne de caractère / même opération que (4) mais en remontant les lignes.

6) "∧" pour désigner la première ligne du fichier

7) "!" pour désigner la dernière ligne du fichier

Avec cette meilleure compréhension de ⟨ligne⟩ et ⟨cible⟩ nous pouvons maintenant examiner des instructions supplémentaires. L'instruction P (RINT) affiche une ou plusieurs lignes. Sa syntaxe est :

⟨ligne⟩ P ⟨cible⟩

Mais pour afficher une seule ligne, la spécification de ladite ligne suffit ; en d'autres termes, ⟨ligne⟩ et ⟨ligne⟩ P ont le même effet.

Revenons à notre fichier :

```
#2P
        2.00 = L'EDITEUR DE TEXTE. BEAUCØUP
#-1
        1.00 = VØICI UN EXEMPLE D'UTILISATIØN DE
#P/APPRENDRE/
        1.00 = VØICI UN EXEMPLE D'UTILISATIØN DE
        2.00 = L'EDITEUR DE TEXTE. BEAUCØUP
        3.00 = D'EXEMPLES NØUS PERMETTRØNT D'APPRENDRE
#! P-3
       13.00 = AU MØINS PØUR L'INSTANT
       12.00 = NØUS VØICI A LA FIN DE CE FICHIER
       11.10 = ENCØRE UNE LIGNE INSEREE
```

```
#-/BBB/P -/123/

        10.00 = BBBBBB
         9.00 = L'EDITEUR EST AMUSANT A UTILISER
         8.20 = AINSI QUE CELLE-CI
         8.10 = C'EST UNE LIGNE INSEREE
         8.00 = TEST 1234

# 12P !

        12.00 = NØUS VØICI A LA FIN DE CE FICHIER
        13.00 = AU MØINS PØUR L'INSTANT

#
```

I.7 L'instruction NEXT

Cette instruction sert principalement à déplacer le pointeur de ligne. Bien qu'elle puisse être utilisée autrement, c'est d'habitude avec la ligne par défaut.

La syntaxe est | N <cible> |

Cette instruction trouve la ligne indiquée par <cible> , l'affiche,et en fait la ligne courante.

Voyons quelques exemples :

```
#^P
        1.00 = VØICI UN EXEMPLE D'UTILISATIØN DE

#N
        2.00 = L'EDITEUR DE TEXTE. BEAUCØUP

#N6
        8.20 = AINSI QUE CELLE-CI

#N-2
        8.00 = TEST 1234

#
```

I .8 Remplacement ou insertion d'une seule ligne

La syntaxe de cette instruction est la suivante :

<ligne> = <texte>

-ligne est spécifiée par son numéro ; il s'agit de la ligne à remplacer ou à insérer ; par défaut c'est la ligne courante.

-texte est le texte de la ligne.

Reprenons notre exemple

#= REMPLACE LA LIGNE CØURANTE ICI

#5.25 CETTE LIGNE EST CREEE AVEC =

#

La première instruction remplace le premier texte de la ligne 8.00, ligne courante.

La deuxième instruction insère une nouvelle ligne de numéro 5.25.

I.9 Modification d'une chaîne de caractères

L'instruction C est utilisée pour remplacer une chaîne de caractères par une autre.

La syntaxe est la suivante :

<ligne> C/ <chaîne>1 / <chaîne>2 / <cible><occurrence>

"/" est le séparateur pour séparer les deux chaînes de caractères : <chaîne>1 est la chaîne à modifier , <chaîne>2 la nouvelle chaîne.

cible détermine l'étendue de la modification.

<occurrence> est l'occurrence de la chaîne 1 dans une ligne.
Si <occurrence> est 1 ou n'est pas spécifiée, seule la première occurrence de chaîne 1 dans toutes les lignes du bloc sera changée.
Si <occurrence> est 2 seule la deuxième occurrence de la chaîne 1 sera modifiée dans les lignes du bloc.
Si <occurrence> est * toutes les occurrences de la chaîne 1 seront modifiées dans toutes les lignes du bloc.

Exemple :

#4 C/ FACILEMENT / AISEMENT/

4.00 = RAPIDEMENT ET AISEMENT SES CARACTERISTIQUES

#8.1 C/VØICI//

8.10 = UNE LIGNE INSEREE

#-5C ; A ; $; SENS ; *

3.00 = D'EXEMPLES NØUS PERMETTRØNT D'$PPRENDRE

4.00 = R$PIDEMENT ET $ISEMENT SES C$R$TERISTIQUES

5.00 = LES LIGNES SUIV$NTES N'ØNT AUCUN SENS

#12C/E/?3/-2.3

12.00 = NØUS VØICI A LA FIN DE CE FICH?ER

11.10 = ENCØRE UNE LIGN? INSEREE

Dans le premier exemple le mot "FACILEMENT" est remplacé par le mot "AISEMENT" .

Dans le deuxième exemple la locution "VØICI" est supprimée dans la ligne 8.10.

Dans le troisième exemple le pointeur de ligne remonte cinq lignes, donc jusqu'à la ligne 3.00 et change chaque occurrence de "A" en "$" dans toutes les lignes jusqu'à la ligne 5.00 qui contient la chaîne de caractères "SENS".

Le dernier exemple montre le changement de la troisième occurrence de "E" en "?" dans la ligne 12.00 et dans la ligne 11.00

I.10 Comment sortir de l'éditeur

Pour sortir de l'éditeur, plusieurs formes sont possibles : STØP, S, ou LØG. Toutes ramènent au FLEX.

I.11 Récapitulatif

Retournons à notre fichier pour illustrer l'effet de ces dernières instructions.

```
# ^P !

 1.00 = VØICI UN EXEMPLE D'UTILISATIØN DE
 2.00 = L'EDITEUR DE TEXTE. BEAUCØUP
 3.00 = EXEMPLES NØUS PERMETTRØNT D'ØPPRENDRE
 4.00 = RØPIDEMENT ET ØISEMENT SES CØRØCTERISTIQUES
 5.00 = LES LIGNES SUIVØNTES N'ØNT ØUCUN SENS
 5.25 = CETTE LIGNE EST CREEE AVEC =
 8.00 = REMPLACE LA LIGNE CØURANTE ICI
 8.10 = UNE LIGNE INSEREE
 8.20 = AINSI QUE CELLE-CI
 9.00 = L'EDITEUR EST AMUSANT A UTILISER !
10.00 = BBBB BBBB
11.00 =
11.10 = ENCØRE UNE LIGN? INSEREE
12.00 = NØUS VØICI A LA FIN DE CE FICHI?R
13.00 = AU MØINS POUR L'INSTANT

# 2C/E /E 6800 / 1 5

 2.00 = L'EDITEUR DE TEXTE 6800. BEAUCØUP

# / BBB /

10.00 = BBBBBBBB
```

```
#- ; VØICI ; C  'E' XX ' !

        1.00 = VØICI UN XXXEMPLE D'UTILISATIØN DE
        2.00 = L'XXDITEUR DE TEXTE 6800. BEAUCØUP
        3.00 = D'XXXEMPLES NOUS PERMETTRØNT D'ØPPRENDRE
        4.00 = RØPIDXXMENT ET ØISEMENT SES CØRØCTERISTIQUES
        5.00 = LXXS LIGNES SUIVØNTES N'ØNT ØUCUN SENS
        5.25 = CXXTE LIGNE EST CREEE AVEC =
        8.00 = RXXMPLACER ICI LA LIGNE PØINTEE
        8.10 = UNXX LIGNE INSEREE
        9.00 = L'XXDITEUR EST AMUSANT A UTILISER
       11.10 = XXCØRE UNE LIGNE INSEREE
       12.00 = NØUS VØICI A LA FIN DXX CE FICHI?R

#N-3
       10.00 = BBBBBBBB
#-1 I
        9.10 = TEST-TEST-TEST
        9.20 = 1234567890
        9.30 = # D !
        BØTTØM ØF FILE REACHED
#I D !
        BØTTØM ØF FILE REACHED
#I P !
#S
```

Ceci n'était qu'une introduction à l'éditeur de texte. Le chapitre suivant décrit chaque instruction en détail avec des exemples. Il est important de lire et étudier le manuel complètement pour parfaitement comprendre l'éditeur et en utiliser pleinement toutes les possibilités.

CHAPITRE II

LES COMMANDES DE L'EDITEUR

Ce chapitre décrit plus précisément tous les ordres d'édition, l'utilisation des aspects particuliers.

La lecture de l'introduction vous a donné une vue générale sur les possibilités de l'éditeur rendant ainsi plus compréhensibles les descriptions détaillées.

Avant d'examiner la description complète des pseudo-instructions de l'éditeur, nous aborderons quelques points généraux.

II.1 Considérations générales

II.1.1 L'utilisation des chaînes

Plusieurs pseudo-instructions de l'éditeur utilisent des chaînes de caractères comme arguments. Ces chaînes remplacent ou se lient à des chaînes du texte sur lequel l'utilisateur travaille.

Une chaîne argument constituée de caractères commence et finit par un séparateur.

Les séparateurs ne sont pas considérés comme faisant partie de la chaîne pour pouvoir être utilisés dans les opérations de remplacement ou de liaison.

Bien que les séparateurs dans les descriptions suivantes soient représentés fréquemment par des slashs, "/", tout caractère non alphanumérique autre que le blanc peut être utilisé comme séparateur :

* / () $ =, . : etc...

Notez que les caractères suivants ne peuvent pas être utilisés pour entourer une chaîne à moins d'être précédés par un signe plus "+" ou moins "-" qui signale que la cible est au-dessus de la ligne courante :

- "∧" qui marque la première ligne du fichier.
- "!" qui marque la dernière ligne du fichier.
- le caractère LINØ,#, utilisé pour signaler les numéros de lignes.

Le caractère séparateur est redéfini à chaque nouvelle demande par sa présence avant une chaîne. Si deux chaînes existent dans une instruction (comme dans l'instruction CHANGE) le même caractère séparateur doit être utilisé pour chaque chaîne.

Toutes les instructions de l'éditeur utilisent l'indication de ligne précédant l'instruction pour positionner le pointeur avant toute exécution de l'instruction.

Le paramètre ⟨ligne⟩ peut bien sûr être absent laissant le pointeur à sa position actuelle.

Tous les caractères suivants peuvent être utilisés comme référence de ligne.

1) Tout entier	Référence à un numéro de ligne donné
2) +N	Signifie la nième ligne suivante
3) -N	Signifie la nième ligne précédente
4) / chaine de caractères /	Renvoie à la prochaine ligne du fichier qui contient la chaine de caractères indiquée
5) -/ chaine de caractères /	Renvoie à la dernière ligne précédente qui contient la chaine de caractères indiquée
6) ∧	Marque la première ligne du fichier
7) !	Marque la dernière ligne du fichier
8) absence	par défaut la ligne pointée.

La plupart des instructions de l'éditeur requièrent une spécification de cible : <cible>. Ce pour indiquer à l'éditeur de traiter toutes les lignes, de la courante à celle spécifiée par <cible>
Par défaut, <cible> est pris à 1 ; en conséquence seule la ligne courante sera concernée par l'instruction.
Tous les caractères suivants peuvent être utilisés comme référence de cible :

1)un entier N	indique que n lignes sont concernées par l'opération (à partir de la courante).
2)# N	indique le N° de la dernière ligne concernée.
3)/ chaîne /	repère la première ligne (après la courante bien sûr) contenant la chaîne en question.
4)- / chaîne /	repère la première ligne au-dessus de la ligne courante contenant la chaîne en question.
5)∧	toutes les lignes sont concernées jusqu'à la première.
6)!	toutes les lignes sont concernées jusqu'à la dernière.
7)+ ou - N	indique que n lignes sont concernées par l'opération et dans quelle direction par rapport à la ligne courante.
8)défaut	la cible est 1 et seule la ligne courante est affectée.

Comme nous l'avons vu, la forme <cible> est utilisée pour indiquer la gamme de lignes auxquelles l'instruction s'applique ; l'instruction s'appliquera à chaque ligne à partir de la ligne spécifiée par <ligne> et continuera jusqu'à atteindre la cible.

Si une chaîne <cible> est spécifiée, l'instruction s'appliquera à toutes les lignes de texte jusqu'à atteindre une ligne contenant la chaîne. La recherche se fait en descendant le fichier sauf si la cible est précédée d'un signe moins, auquel cas la recherche (et le traitement) se font en remontant le fichier.

Les cibles peuvent être également précédées d'un signe + (recherche descendante). Si un numéro de ligne cible est spécifié, le traitement commence à <ligne>et s'achève à la ligne cible. Voici quelques exemples de <cible>:

2

+ 10

- 3

/ CHAINE /

+ / CIBLE CHAINE /

- / DEPLACEMENT ARRIERE A UNE CHAINE /

+ * TØUT SEPARATEUR CØNVIENT PØUR UNE CHAINE *

+ + MEME LE SIGNE PLUS CØNVIENT +

23.00

II.1.2 <u>Spécification d'un numéro de colonne</u>

Toute indication "/ chaîne /"peut être suivie d'un numéro de colonne immédiatement après le second séparateur pour indiquer que la chaîne en question doit commencer dans la colonne spécifiée pour être trouvée.

Si la colonne spécifiée n'est pas dans l'ensemble de la zone affectée, la demande sera ignorée.

Voici quelques exemples :

/ IDENTIFICATEUR/11
/ PRØGRAMME /77
* ETIQUETTE * 2
$ CØMMENTAIRE $ 30

II.1.3 Caractère "ordre répété"

Un caractère "ordre répété" a été introduit dans l'éditeur pour permettre de répéter exactement le dernier ordre dans le tampon d'entrée.

Si un ordre entraîne une erreur ou change le contenu du tampon d'entrée, le commentaire ILLEGAL CØMMAND (ordre illégal) sera affiché à chaque utilisation du caractère de répétition jusqu'à ce qu'un autre ordre répétable soit donné.

Le caractère de répétition est CTRL R (12_{hex} ASCII).

Quelques exemples d'ordres qu'il peut être utile de répéter :

PRINT 15	pour imprimer d'un coup un écran de lignes.
NEXT	pour dérouler pas à pas un fichier avec une seule touche.
ΛCØ !!	pour remplir rapidement l'espace de travail.
FIND/Chaîne de caractères/	si la première chaîne trouvée n'est pas la bonne.

II.1.4 Utilisation du caractère de fin de ligne (EØL)

L'éditeur comprend un caractère EØL (End of Line) pour permettre des ordres multiples dans une seule ligne.

Les ordres INSERT et ØVERLAY sont des exceptions car ils ne peuvent pas être suivis par d'autres ordres.

Le caractère EØL peut être changé de façon interactive par l'ordre SET.

Exemple de l'utilisation de EØL (avec "$" comme caractère EØL):

Λ D 2 $ P 10 $ T

La séquence supprimera les deux premières lignes du fichier, imprimera les dix lignes suivantes et finalement renverra le pointeur au début du fichier.

II.1.5 Utilisation du tabulateur

L'utilisateur peut spécifier interactivement un caractère de tabulation, et ceci jusqu'à 20 arrêts de tabulation par ligne.

Le caractère de tabulation peut être inséré dans une ligne où il sera pris en considération après envoi du caractère de fin de ligne.

Si les arrêts ou le caractère de tabulation n'ont pas été spécifiés auparavant, mais qu'un caractère a servi de tabulateur dans tout le fichier, il peut encore être pris en considération en l'affectant à TAB, plaçant les arrêts et utilisant l'ordre EXPAND sur le fichier.

Notons que si un caractère de tabulation a été spécifié il sera automatiquement pris en considération par les ordres d'insertion INSERT ou de remplacement REMPLACE.

Si toutefois un caractère de tabulation est ajouté à l'aide des ordres CHANGE, APPEND ou ØVERLAY il ne sera pas pris en considération jusqu'à ce qu'un ordre EXPAND soit appliqué à la ligne contenant ce caractère de tabulation.

II.2. Instructions de l'éditeur

Il existe quatre groupes d'instructions :

- les instructions d'environnement
- les instructions systèmes
- les déplacements du pointeur de ligne
- les instructions d'édition

Une description complète de toutes les instructions de chaque groupe est donnée ci-après.

Dans les descriptions suivantes, les caractères compris entre parenthèses (   ) sont optionnels et peuvent être omis.

Un slash (/) est utilisé pour séparer les options.

II.2.1 Les instructions d'environnement

H	(EADER)	(<NØMBRE>)

Signification :

Cette instruction permet d'afficher la ligne suivante :

12345678901234567890......

sur 80 colonnes ou le nombre de colonnes précisé, pour faciliter la mise en page du texte en indiquant les colonnes dix par dix.

Les colonnes qui contiennent un arrêt de tabulation ont un caractère "-" à la place du chiffre.

Si le nombre de colonnes est précisé, il sera ensuite gardé comme valeur par défaut, lors de la frappe de H.

Exemples :

HEADER 72	affiche une ligne de 72 colonnes
H 30	affiche une ligne de 30 colonnes.

NU(MBERS) (ØFF/ØN)

Signification :

Le drapeau numéro de ligne est arboré ou amené.

Si le drapeau est amené les numéros de ligne ne seront jamais imprimés.

Si ni ØFF ni ØN sont spécifiés, le drapeau sera inversé par rapport à l'état actuel.

Exemples :

NUMBERS ØFF	n'imprimera pas les numéros de ligne
NU ØN	les numéros de ligne seront imprimés
NU	bascule de ØN sur ØFF et de ØFF sur ØN

REN(UMBER)

Signification :

La pseudo-instruction renumérote toutes les lignes dans le fichier en cours d'édition.

La première ligne du fichier renuméroté sera 1.00 ; le pas est de 1.

La ligne courante avant l'ordre sera encore la ligne courante après l'ordre (mais son numéro sera probablement différent).

Exemples :

RENUMBER	renumérote les lignes dans le fichier de travail en cours
REN	" "

SET<nom> = '<caractère>'

Signification :

SET est utilisé pour définir certains caractères ou symboles spéciaux. Les noms peuvent être :

TAB   Caractère de tabulation

FILL   Caractère de remplissage de la tabulation

EØL   Caractère de fin de ligne qui peut être utilisé pour séparer plusieurs ordres dans une seule ligne

LINØ   Caractère de drapeau de numéro de ligne qui est utilisé pour indiquer qu'une cible est un numéro de ligne spécifique.

Les valeurs implicites sont :

TAB et EØL	absence
FILL	'espace'
LINØ	'#'

Exemples :

SET TAB='/'	Le caractère de tabulation est un slash
SET TAB=	Invalide la tabulation
SET FILL = ' '	Le caractère de remplissage de la tabulation est un espace
SET EØL = '$'	Le caractère de fin de ligne est $
SET LINØ= '*'	Le caractère de drapeau de numéro de ligne est *

TAB (<colonnes>)

Signification :

TAB est utilisé pour placer les arrêts de tabulation : les précédents arrêts sont effacés.

Si aucune colonne n'est spécifiée, la seule action est d'effacer tous les arrêts de tabulation.

Tous les caractères de tabulation qui sont au-delà du dernier arrêt de tabulation sont laissés dans le texte.

Le nombre maximum d'arrêtsde tabulation est vingt.

Les arrêts de tabulation doivent être saisis en ordre ascendant.

Exemples :

TAB 11, 18, 30	met des arrêts de tabulation aux colonnes 11, 18, 30
TAB 7, 72	met des arrêts de tabulation pour un programme FORTRAN
TAB	nettoie tous les arrêts de tabulation

V (ERIFY) (ØN/ØFF)

Signification :

Le drapeau de vérification est arboré ou amené.

Ce drapeau est utilisé dans les instructions CHANGE et NEXT (et plusieurs autres) pour visualiser leurs résultats.

Si ni "ØN" ni "ØFF" ne sont spécifiés, le drapeau sera inversé par rapport à son état initial.

On contrôle l'état de VERIFY en constatant si dans les instructions concernées les résultats s'affichent ou non.

Exemples :

VERIFY ØFF	ote la vérification
V ØN	met la vérification
V	inverse ØN et ØFF

X

Signification :

"X" est l'ordre commande du curseur.

A chaque entrée de cet ordre, l'éditeur renvoie la chaîne de six caractères spéciaux précédemment définie.

Ceci est utile pour des applications spéciales.

Exemple :

X   sort la chaîne de commande du curseur

Z (ØNE) (C1,C2)

Signification :

ZØNE est utilisé pour restreindre toutes les recherches de sous-chaines (FIND, CHANGE, cible , etc..) seulement entre les colonnes C1 et C2 (comprises).

Toutes les sous-chaînes commençant en dehors de ces colonnes ne seront pas détectées.

Si C1 et C2 ne sont pas spécifiés, les zones seront prises par défaut (colonnes 1 et 136).

Exemple :

ZØNE 11, 29	restreint les recherches entre les colonnes 11 à 29.
ZØNE	recherche entre les colonnes 1 et 136.

II.2.2 Les pseudo-instructions "système"

LØG

Signification :

sortie de l'Editeur

S (TØP)

Signification :

même que LØG

II.2.3 Déplacements du pointeur de ligne

B(ØTTØM)

Signification :

BØTTØM déplace le pointeur jusqu'à la dernière ligne du fichier, qui devient la ligne courante.

Exemple:

BØTTØM	La dernière ligne du fichier devient ligne pointée
B	" "

F(IND) <cible> (occurrence)

Signification :

FIND déplace le pointeur de la ligne courante jusqu'à la ligne spécifiée dans la cible, qui devient alors la ligne courante.

Si le drapeau de vérification VERIFY est arboré (voir VERIFY), la ligne sera imprimée. Si l' occurrence est spécifiée (un entier ou un astérisque) la commande sera répétée le nombre de fois indiqué. Si occurrence est un entier, frapper un blanc entre le second séparateur de la chaine et l' occurrence (cf exemple 2) pour éviter la confusion avec un suffixe de chaine (cf exemple 3).

Un astérisque entraine la recherche jusqu'à la dernière ou la première ligne du fichier (cf exemple 4-5).

Si la cible n'est pas trouvée, le pointeur ne sera pas déplacé.

Exemple :

FIND/RENARD/	Recherche la prochaine ligne qui contient la chaine "RENARD"
F/GØUPIL/ 3	Recherche les trois prochaines lignes qui contiennent la chaine /GØUPIL/
F/GØUPIL/30	Recherche dans la ligne pointée la chaine GØUPIL commençant en colonne 30
F/GØUPIL/ *	Recherche toutes les occurrences de la chaine "GØUPIL" jusqu'à la fin du fichier.
F/GØUPIL/7 *	Recherche toutes les lignes précédentes qui ont le mot "GØUPIL" commençant en colonne 7 jusqu'au début du fichier.

N(EXT) ((<cible> (<occurence>))

Signification :

La ligne spécifiée par la cible devient la ligne courante. Si le drapeau de vérification VERIFY est arboré,cette ligne sera imprimée. Si l' occurrence est spécifiée (nombre entier),elle indique laquelle parmi les lignes qui contiennent la cible devient la ligne courante.

Si la cible n'est pas trouvée, le pointeur de ligne sera déplacé à la fin du fichier (ou au début si le déplacement est ascendant).
Si aucune cible n'est mentionnée la ligne suivante devient la ligne courante.

Exemples :

NEXT 5	Le pointeur de ligne est déplacé à la cinquième ligne suivante
N	La ligne suivante devient ligne courante.
N-10	La dixième ligne précédente devient ligne courante.
N/GØUPIL/	La prochaine ligne qui contient la chaine "GØUPIL" devient ligne courante.
N/GØUPIL/ 3	La troisième ligne contenant la chaine "GØUPIL" devient ligne pointée.

T(ØP)

Signification :

La première ligne du fichier devient la ligne courante.

Exemple :

TØP	Met le pointeur de ligne à la première ligne du fichier.

II.2.4. Les instructions d'édition

A (PPEND)/ <chaîne> /(<cible>)

Signification :

APPEND lie la chaine spécifiée au dernier caractère de la ligne courante et des lignes suivantes jusqu'à ce que la cible soit atteinte. Si la chaine de caractères est postfixée par un numéro de colonne, le début de la chaine est inséré à la colonne spécifiée et non à la fin de la ligne. Tous les caractères qui suivaient la colonne spécifiée dans l'ancienne ligne sont effacés.

Exemples :

APPEND/./	Ajoute un point à la fin de la ligne courante
A *HELLØ*2	Ajoute le mot "HELLØ" à la fin de la ligne courante et à la fin de la ligne suivante.
A/GØUPIL/73*FIN*7	Ajoute le mot "GØUPIL" à partir de la ligne courante et des lignes suivantes jusqu'à trouver une ligne contenant la chaine FIN commençant à la 7ème colonne.

C(HANGE)/<chaine 1>/<chaine 2>/(<cible> (<occurrence>))

Signification :

Remplace la chaine de caractères1 par la chaine de caractères2

Si aucune cible n'est spécifiée, seule la ligne courante est modifiée.Les slashes "/" représentent tout caractère séparateur autre qu'un blanc.

<occurrence> spécifie quelle occurrence de la chaine 1 doit être remplacée dans chaque ligne : c'est soit un entier soit un astérisque ('*') qui signifie que toutes les occurrences de la chaine 1 doivent être remplacées par la chaine 2 . Par défaut seule la première occurrence sera modifiée.

NOTE :

Si l'occurrence est spécifiée et si la modification ne doit avoir lieu que sur la ligne courante alors la cible doit être un 1 (un).

Exemples:

CHANGE/CELUI/CELLE/	Remplace le premier "CELUI" de la ligne courante par "CELLE".
C/A/B/1*	Change tous les "A" par "B" dans la ligne courante.

C/PREMIER/DERNIER/10	Change le premier "PREMIER" par "DERNIER" dans la ligne courante et dans les neuf lignes suivantes.
C/NØUVEAU/ANCIEN/UN ØBJET/	Change le premier "NØUVEAU" par "ANCIEN" dans chaque ligne jusqu'à ce qu'une ligne contenant la chaine "UN OBJET" soit rencontrée.
C,A,, -10*	Enlève tous les "A" dans la ligne pointée et dans les neuf lignes précédentes.
C*HELLØ*	Efface la chaine de caractères "HELLØ" dans la ligne courante.

CØ(PY) (<cible-destination> (<cible-gamme>))

Signification :

La ligne courante et les lignes suivantes jusqu'à la cible-gamme sont copiées après la cible-destination.

La destination par défaut est 1 : la ligne courante est alors copiée après la ligne suivante.

La gamme par défaut est 1 : dans ce cas seule la ligne courante est copiée.

Après l'éxécution de la pseudo-instruction, le pointeur de ligne se place à la nouvelle position de la dernière ligne copiée. Il peut y avoir renumérotation de certaines lignes sans qu'un message de renumérotation apparaisse.

Exemples :

CØ//18	Copie de la ligne courante après la ligne 18.
CØ//34	Copie de la ligne suivante et les trois lignes suivantes après la ligne 3.
CØ/ØRDRE/+/RANG	Copie de la ligne courante et des lignes suivantes jusqu'à la ligne contenant la chaine de caractère "RANG" après la prochaine ligne qui contient la chaine de caractère "RANG".

D(ELETE) (<cible>)

Signification :

La ligne courante (et les lignes suivantes jusqu'à la ligne de cible sont détruites). Après l'éxécution de l'instruction. la ligne courante sera la ligne suivant la dernière ligne détruite)

Exemple :

DELETE 5	Suppression de cinq lignes (la ligne courante et les quatre lignes suivantes)
D	Suppression de la ligne courante
D/CHAINE/	Suppression de la ligne courante jusqu'à la prochaine ligne qui contient la chaîne de caractères "CHAINE".

EXP (AND) (< cible >)

Signification :

Le caractère de tabulation spécifié est étendu à toutes les lignes depuis la ligne courante jusqu'à la ligne cible.

Les caractères de tabulation sont normalement étendus lorsque des lignes sont insérées dans le fichier mais cette commande est d'une grande utilité lorsqu'on a oublié de définir un caractère de tabulation.

Exemple :

EXPAND 100	extension à 100 lignes à partir de la ligne courante.
EXP	extension à la ligne courante.

(I)NSERT

Signification :

L'éditeur entre les données écrites dans la mémoire tampon, en indiquant les numéros de lignes et insère les lignes après la ligne courante (si ces numéros ne sont plus valables voir la commande NUMBERS).

La saisie dans le tampon continue jusqu'à ce que le caractère d'arrêt (le dièse "#") soit rencontré au début d'une ligne (dans la première colonne).

Les caractères qui suivent ce caractère d'arrêt sont traités comme une commande d'édition.

L'éditeur essaiera de choisir le pas pour pouvoir insérer 10 lignes au moins. Si ce n'est pas possible, il prendra le plus petit pas possible.

Le pointeur de ligne sera positionné à la dernière ligne insérée.

L'éditeur peut renuméroter les lignes de textes qui suivent le texte inséré si les numéros des lignes insérées chevauchent les numéros de lignes existantes dans le fichier.

Exemple :

INSERT ou I	Accepte la saisie de nouvelle ligne après la ligne pointée.

I(NSERT) 〈 texte 〉

Signification :

Le texte qui suit immédiatement le caractère de séparation (un blanc) après la commande (I) sera écrit sur une ligne après la ligne courante,cette nouvelle ligne devient alors la ligne courante.

L'éditeur peut renuméroter les lignes de textes qui suivent le texte inséré si le numéro de la ligne insérée coïncide avec le numéro de lignes existantes dans le fichier.

Exemple :

```
2 = FIN
3 =
I   RAJØUTER
```

MØ(VE) (<destination>(<cible-gamme>))

Signification :

La ligne courante et les lignes suivantes jusqu'à la ligne cible (gamme)sont récrites. après la ligne destination.

La destination implicite est 1, c'est-à-dire que la ligne courante est récrite au-dessous de la ligne suivante du fichier. La gamme implicite est 1 : une seule ligne est récrite.

Après exécution de la commande le pointeur de ligne sera placé à la dernière ligne récrite. Quelques lignes peuvent être renumérotées après un mouvement sans qu'un message de renumérotation soit exprimé.

Exemple :

MØVE 3	Récrit la ligne courante trois lignes plus bas
MØVE-/PRØGRAMME/5	Récrit la ligne courante et les quatre suivantes après la première ligne précédente du fichier qui contient la chaine de caractère "PRØGRAMME"
MØ 10 - 5	Récrit la ligne courante et les quatre lignes précédentes au-dessous de la ligne 10 du fichier.

Ø(VERLAY) (< séparateur>)

Signification :

La ligne courante est imprimée et une ligne de saisie est acceptée ensuite sur le terminal : la ligne de recouvrement.

Cette ligne est positionnée directement au-dessous de la ligne courante.

Chaque caractère de recouvrement qui est différent du caractère séparateur (un blanc par défaut) remplacera le caractère correspondant dans la ligne courante.

La ligne modifiée sera imprimée si VERIFY est ØN.

Exemple :

```
# ØVERLAY
                  =   BØICI LA PØNNE VIGNE
   ØVERLAY            V        B     L
                  =   VØICI LA BØNNE LIGNE
```

Ø(VERLAY)<d><texte>

Signification :

Cette instruction est semblable à la commande de recouvrement précédente ØVERLAY mais la ligne courante n'est pas imprimée, le texte qui suit le caractère délimiteur est considéré comme recouvrement supplémentaire.

Exemple :

```
ØVERLAY-------LA------------ ECRITE

25.00 = VØILA LA BØNNE LIGNE ECRITE
```

P(RINT) (<cible>)

Signification :

Impression de la ligne courante et des lignes suivantes jusqu'à la ligne cible. Par défaut, seule la ligne courante est imprimée.

Exemples :

P	Imprime la ligne pointée
PRINT 5	Imprime la ligne pointée et les quatre suivantes
P-10	Imprime la ligne pointée et les neuf précédentes
100P	Imprime la ligne 100
PRINT*CHAINE*	Imprime toutes lignes précédentes jusqu'à la première ligne contenant la chaine de caractère CHAINE.

NOTE : Un retour chariot imprime la ligne courante.

R(EPLACE) (<cible>)

Signification :

Suppression de la ligne courante jusqu'à la ligne cible.

L'éditeur se met en mode saisie dans mémoire tampon ; les nouvelles lignes sont mises dans l'espace libéré. Il n'est pas obligatoire de rentrer autant de lignes que de lignes supprimées. Les numéros deslignes insérées peuvent être différents de ceux des lignes supprimées. Le pointeur de ligne sera positionné à la dernière ligne saisie.

Par défaut seule la ligne courante est supprimée.

Exemple :

R	Remplace la ligne courante.
REPLACE 10	Remplace 10 lignes (ligne courante comprise)
R/GØUPIL/	Remplace toutes les lignes à partir de la ligne courante jusqu'à la ligne qui contient la chaîne de caractères"GØUPIL".

= ⟨texte⟩

Signification :

L'instruction"=" remplace la ligne pointée par le texte exprimé.

Le texte de remplacement commence avec le premier caractère qui suit le signe =.

Le pointeur de ligne ne bouge pas.

Exemple :

```
=   C'EST UN NØUVEAU TEXTE
```

CHAPITRE III

MANIPULATION DE FICHIERS

Ce chapitre décrit comment créer et accéder à des fichiers de textes, comment manipuler ces fichiers et comment sortir de l'éditeur.

III-1 - Description générale

La syntaxe générale de la commande d'exécution de l'éditeur est :

EDIT, fichier 1 (, fichier 2)

L'extension implicite est .TXT et le lecteur de disque implicite est le lecteur du disque de travail.

Si on spécifie seulement 'fichier 1' et que le nom de ce fichier n'existe pas sur le disque, un nouveau fichier est créé à ce nom.

Lors de la création de nouveaux fichiers, l'éditeur envoie le message suivant :

NEW FILE

1.00 =

Si la ligne de commande EDIT ne spécifie que 'fichier 1' et que ce nom existe déjà sur le disque, ce fichier sera chargé dans le tampon de saisie et l'éditeur imprimera un '#' sur le terminal, signifiant qu'il est prêt à accepter des ordres de l'utilisateur.

Quand le processus d'édition est terminé, le fichier original garde son nom mais il a maintenant comme extension BAK pour "back up".

Si un fichier .BAK du même nom existe déjà l'éditeur demande :

DELETE BACK UP FILE ? (Y - N)

Détruire le fichier secondaire (ØUI ou NØN).

Une réponse positive (Y) (ØUI) détruit l'ancien fichier .BAK est en crée un nouveau. Une autre réponse renvoie à FLEX.
Le nouveau fichier aura le même nom que le précédent y compris son extension.

La dernier forme de EDIT permet d'assigner au nouveau fichier un nom spécifique différent de celui du fichier qui conserve alors le sien.

Notons que lorsque on édite un fichier existant, un nouveau fichier est toujours créé, l'original restant intact même si son nom peut être changé.

Plusieurs exemples nous aideront à clarifier cette syntaxe :

1) Si vous voulez créer un fichier appelé TEST.TXT (aucun fichier de ce nom n'existe encore sur le disque). Il faut taper la commande suivante : EDIT, TEST

L'éditeur répond :
NEW FILE
1.00=

Il est prêt à accepter des lignes de texte.

2) Vous avez créé un fichier TEST.TXT et vous voulez l'éditer pour le modifier.

Il faut taper la commande suivante :

EDIT, TEST

Le fichier TEST.TXT est alors chargé en mémoire et l'éditeur est prêt à accepter des ordres.

A la fin de la modification,la main renvoyée au FLEX, le premier fichier TEST.TXT s'appelle maintenant (mais son contenu reste inchangé) TEST.BAK et le fichier modifié s'appelle TEST.TXT.

Si de nouvelles modifications doivent être faites la même procédure peut être employée :

EDIT, TEST

L'éditeur vous demande alors si vous voulez détruire le fichier "TEST.BAK" ; si oui, la même procédure s'applique, un nouveau fichier .BAK sera créé prenant la version en cours avant les nouvelles modifications et ainsi de suite...

3) Si vous voulez éditer un fichier mais lui donner un nouveau nom vous devez taper la commande suivante :

EDIT, TEST,TEST2

Le fichier TEST.TXT sera chargé en mémoire mais le nouveau fichier s'appelera TEST 2.TXT. Cette forme de commande est aussi utilisée quand il s'agit d'éditer un fichier à partir d'un lecteur et charger le nouveau fichier sur un autre lecteur, par exemple :

EDIT, 0.TEST, 1.TEST

Le fichier TEST.TXT du lecteur 0 sera édité et le nouveau fichier TEST.TXT sera copié sur le lecteur 1 (mais il ne faut pas que TEST.TXT existe déjà sur le lecteur 1)

Une fois dans l'éditeur, toutes les commandes d'édition s'appliquent à la version FLEX de l'éditeur à part quelques exceptions expliquées ci-dessous.

III-2 - Sortie de l'éditeur

Il faut utiliser la commande STØP ou "S". On peut aussi utiliser la commande LØG. Après que LØG, STØP ou S ont été frappés, l'éditeur terminera automatiquement le transfert de l'ancien vers le nouveau fichier disque. Pour l'édition d'un fichier long : la manipulation doit avoir le temps de s'effectuer et les +++ du FLEX n'apparaitront qu'à la fin de l'opération.

III-3 - L'ordre "NEW"

C'est une commande additionnelle de la version FLEX de l'éditeur. Cette commande aide l'usager dans la manipulation de fichiers de textes plus importants que ceux que peut contenir la mémoire centrale. Lors de l'édition d'un tel fichier l'éditeur ne chargera en mémoire que ce qu'il peut.

La commande NEW dit à l'éditeur que vous avez fini l'édition de cette partie et que voulez en charger le morceau suivant.

Elle agit de la façon suivante : après avoir frappé NEW, toute la partie du fichier contenu dans la mémoire tampon jusqu'à la dernière ligne pointée est rangée sur le fichier disque .BAK sauf la dernière ligne pointée. Ensuite la plus grande partie possible du fichier original qui n'a pas encore été lue est copiée dans la mémoire tampon. La main sera rendue à l'éditeur et la dernière ligne pointée deviendra la première ligne pointée de cette nouvelle partie.

La commande NEW peut être utilisée aussi souvent que nécessaire mais il faut se souvenir que lorsqu'une portion de fichier a été rechargée sur disque elle n'est plus accessible par l'éditeur. Car il ne peut travailler que sur le texte chargé dans la mémoire tampon ; en conséquence des commandes globales comme CHANGE ou FIND ne peuvent agir que dans le cadre du texte chargé dans la mémoire tampon et non sur le fichier complet, à moins, bien sûr qu'il ne soit compris tout entier dans la mémoire tampon.

L'ordre NEW peut aussi servir lors de la création d'un fichier ; il est en effet possible d'emplir complètement le tampon (l'éditeur délivre alors le message NØT ENØUGH RØØM - pas assez de place).

Frapper alors NEW et tout le début du fichier jusqu'à la ligne courante sera rangé sur disque, libérant ainsi l'espace dans la mémoire tampon. Il n'y aura aucune lecture à partir du disque puisque le fichier vient d'être créé.

III-4 - Taille de la mémoire tampon

Le volume disponible en mémoire tampon est directement proportionnel au volume de mémoire total de l'ordinateur.

Plus la mémoire totale est grande plus le tampon de l'éditeur est important.

L'éditeur s'adapte automatiquement à la taille de la mémoire.

Par exemple dans un GOUPIL 2 en configuration 48Ko, il reste environ 34Ko pour la mémoire tampon (pour le texte donc sur lequel l'utilisateur travaille).

De même, il reste 42Ko dans un GOUPIL 2 en configuration 56Ko.

ANNEXE

1. Les caractères 'système'

Le caractère prêt (prompt) est rangé à l'adresse 0528_{16} ; c'est un dièse # (23_{16} ASCII).

Le caractère de suppression de ligne est celui du FLEX (touche CTRLX) de même que le caractère de suppression du dernier caractère affiché (touche CTRL H ou ←).

La cloche retentit quand le tampon d'entrée est saturé (plus de 136 caractères frappés par ligne).

Le caractère de répétition d'ordre est rangé à l'adresse 0530_{16} ; c'est un CTRL R (12_{16} ASCII) (à ne pas confondre avec la touche REPEAT)

2. Caractéristiques du système

Le plus grand numéro de ligne est 9999.99; si plus de 9999 sont frappées, le compteur de lignes revient à 0 ; en conséquence l'éditeur ne peut être utilisé avec des fichiers de 10000 lignes ou plus (ceci n'est pas vraiment une limitation puisque 10000 lignes vierges occupent 40 Ko de mémoire !).

Quand un numéro de ligne est inférieur à 1, il est nécessaire de faire précéder le point d'un 0 : 0.10 et non .10 par exemple. Ceci pour que le point ne soit pas pris pour un séparateur.

Le tampon d'entrée accepte 136 caractères ; au-delà les caractères sont ignorés ; pour sortir de la ligne il faut revenir (CTRL H ou ←) avant de faire un "retour chariot".

Il y a plusieurs façons d'insérer une ligne au début d'un fichier qui contient déjà une ligne 0.01. L'une est de renuméroter le fichier, 0.01 devenant 1.00 et permettant l'insertion d'une ligne 0.10. Une autre façon de faire est de frapper "INSERT - <x> " où x est le nombre de ligne qui renvoie une ligne au dessus du fichier. Si par exemple la ligne courante est 0.01, la commande "INSERT-1" permet d'insérer avant cette ligne.

3. Erreurs

Les erreurs,qui ne peuvent se produire que rarement,entraînent l'affichage de messages d'erreur très simples imprimés en clair en langue anglaise.

LISTE DES COMMANDES ET ORDRES DE L'EDITEUR DE TEXTE

LISTE DES COMMANDES ET ORDRES DE L'EDITEUR DE TEXTE (suite)

Imprimé en France. — Imprimerie JOUVE, 18, rue Saint-Denis, 75001 PARIS S M T